Supporting communication in Information Centric Networks using the Location/ID Split Protocol and Time Released Caching

Mahdi Aiash, Glenford Mapp
School of Science and Technology
Middlesex University
London, UK
Email: M.Aiash, G.Mapp@mdx.ac.uk

Mohammad Muneer Kallash
School of Engineering
University of Kalamoon
Syrian Arab Republic
Email: M.Kallash@hotmail.com

***Abstract*—The vast majority of current Internet usage is data retrieval and information exchange. As a result, the focus has been shifted from the current location-based system to an Information-Centric system, where information can be cached and accessed from anywhere within the network rather than from the end hosts only. To support this functionality, data must be uniquely identified regardless of the location. Current research efforts in the area of Information-Centric Networks presume the existence of a Convergence Layer protocol that facilitates the functionalities of forwarding, while data caching takes place on a higher-plane. Therefore, this paper proposes a convergence layer protocol, based on the Location/ID Separation Protocol which uses two numbering spaces for data. Unlike other Information Centric architectures in the literature, the proposed approach introduces new procedures to deal with in-network data caching and forwarding separately.**

***Keywords*-Information Centric Networks; Location/ID Split Protocol; Future Internet Architecture; Time Released Caching;**

I. INTRODUCTION

Communication in the current Internet is based on the Client-Server model, where servers share their resources and offer services to clients. In this model, communicating entities exchange the information among themselves. However, new trends in communication systems place more attention on WHAT data are being exchanged rather than WHO are exchanging them [1]. This led to a new communication model, referred to as Information (or Data)-Centric Networking (ICN). The principal paradigm in this model is not an end-to-end communication between hosts. Instead, an increasing demand for highly scalable and efficient distribution of content has motivated the development of architectures such as [2]–[4] that focus on information objects, their properties, and receiver interest in the network to achieve efficient and reliable distribution of such objects [5].

The main reason for advocating the departure to the information-centric model is that the current Internet is mostly used for content access and delivery, with a high volume of digital content delivered to users who are only interested in the actual content rather than the source location [6]. In this sense, content names are decoupled from hosts or servers addresses. So unlike current IP-based addresses which use a single numbering system to identify hosts and define their locations, the ICN separates the roles of identifier and locator, which implies that each data object will be identified, using a unique name called Named Data Object (NDO) without being mapped to a specific location. This will lead to one of the salient features of the ICN which is application-independent caching of contents, where network elements like routers will be able to cache recent contents and resend them when requested by other end-users.

Obviously, current addressing and routing schemes such as IPv4/IPv6 and Border Gateway Protocol (BGB) [7] cannot support this functionality of identifying and routing data regardless of their location. Hence, we strongly believe in the need for new protocol (and system) that acts as a Convergence Layer (CL) protocol to support the new functionalities required in the ICNs. In this regards, the concept of ID/Location Split which separates the roles of identifiers and locators has been used by a number of newly proposed addressing schemes such as the Identifier-Locator Network Protocol (ILNP), Y-Comm Address scheme and Location/ID Split Protocol (LISP) [8]–[10]. The author believes that the concept of the ID/Location split complies with the features of separating data identifier from its location in the ICN, and hence, the well-developed LISP network architecture could be used to as a CL to support communication in ICN. The anticipated benefits of using the LISP are as follows:

1) The LISP has already developed a network infrastructure along with the networking elements and services. These could be used to support ICN communication without the need for introducing new entities and services.
2) Since addressing in LISP is based on IPv4/IPv6, there is no need to change the current networking elements in the Internet. Furthermore, LISP-capable routers are fully compatible with non-LISP capable routers.
3) The transactions between the LISP's network elements have been defined using newly-developed packets (called LISP-Packets). These packets will facilitate

data routing in ICN.

4) The LISP working group has proposed mechanisms for security and enhancing Quality of Service (QoS). Deploying the LISP with ICN, will enable us to benefit from these mechanisms.
5) The LISP protocol has been recently implemented in the new Cisco Nexus 7000 Series switches [13], this will enable us to build a real lab for testing the proposed model in this paper.

Due to these reasons, this paper investigates how to use the LISP to support addressing and routing in ICN.

Although, the communication in ICNs is based on in-network caching and efficient routing/forwarding of data, most of the proposed architectures for ICNs did not deal with these two functionalities separately. As will be described in section II, most of the ICN architectures in the literature assume data to be cached and then routed by intermediate network elements such as routers. This places too much responsibilities on the networking elements and relies on legacy mechanisms such as current routing protocols to provide the required functionalities. The authors believe that for communication in ICNs, data caching and forwarding must be addressed separately, which means the need for new caching and forwarding procedures. Therefore, in addition to the novel network structure -based on the LISP-, the paper introduces a potential procedure for data caching and forwarding.

The rest of this paper is organized as follows. Section 2 discusses design choices and features of a number of proposed ICN architectures. Section 3 gives an overview of the LISP network architecture and its basic operation. Section 4 introduces the proposed model, it describes how implementing LISP could support data retrieval and routing in ICN. Section 5 describes the proposed caching and forwarding procedures. A conclusion is given in Section 6.

II. Overview of Information Centric Network Architectures

This section illustrates few of the most-known ICN approaches at a high level with the purpose of providing a general understanding of them.

A. *Data-Oriented Network Architecture (DONA)*

The DONA relies on a new class of network entities called resolution handlers (RHs). Name resolution is accomplished through the use of two basic functions: Register and Find [2]. As shown in Fig 1, initially, nodes that are authorized to act as data sources send Register packets to register their NDOs with the local RH. Each RH will maintain a registration table that maps a name to both a next-hop RH and the distance to the copy (in terms of the number of RH hops, or some other metric). When a client request a specific data (identified by a unique NDO), it sends a Find packet to the local RH, when a FIND arrives, the forwarding rule is straightforward: if there is an entry in the registration table, the FIND is sent to the next-hop RH (and if there is more than one, the choice is based on the local policy and which entry is closest). Once a copy of the data is found, it will be returned through the reverse RH path.

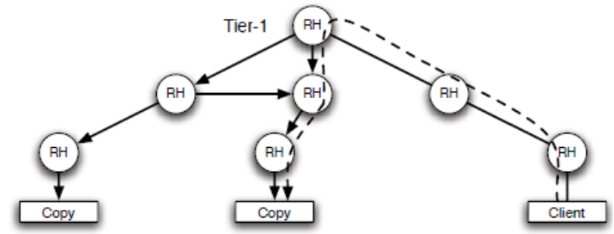

Figure 1. The Registration Stage (solid arrows). Routed Data (dashed arrows) [2]

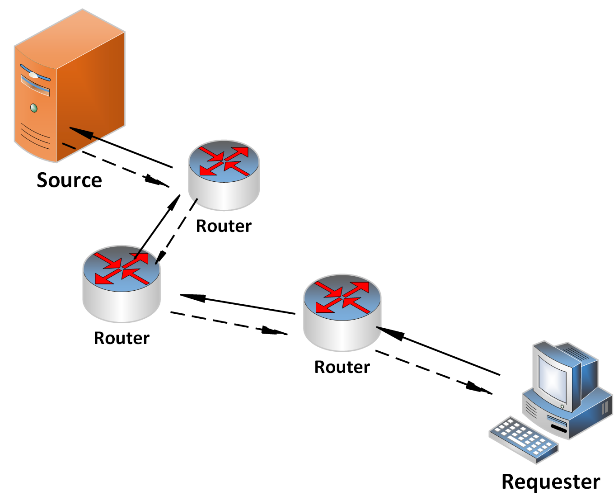

Figure 2. The Request (solid arrows). Forwarded Data (dashed arrows)

B. *Content-Centric Networking (CCN)*

In CCN, NDOs are published at nodes, and routing protocols are used to distribute information about the NDOs location. Communication is initiated by issuing a request messages (referred to as Interest). An Interest is routed by a data name instead of a host identifier. Because the data name has a hierarchical structure like a file system pathname such as /CCN.org/cnlab/ccnpaper', each CCN router can forward an Interest in a hop-by-hop manner [3]. A CCN router maintains a pending interest table (PIT) for outstanding requests, this enables request aggregation as a CCN router will not forward a second request for the same NDO. Once a copy of the request data is found, it will be routed back on the reverse request path, as shown in Fig 2.

C. *Publish-Subscribe Internet Routing Paradigm (RSIRP)*

This approach is based on the concept of publish/Subscribe (Pub/Sub) model, where hosts can join a network, publish data, and subscribe to publications. However, when a node publishes data, no data transfer actually takes place. Only when a node subscribes to a named piece of data, the network finds the publication and creates a delivery path from the publisher to the subscriber [4]. The network architecture is composed of three modules: Rendezvous module which is a distributed database that maps the wanted data to the subscriber. The Forwarding module, which is used to deliver data from one location to another, the forwarding procedure is based on label switching as each packet will have a label that will help the router to decide on the next hop to forward the packet. The Topology module creates and maintains delivery trees used for forwarding traffic accomplished by the forwarding module. A node publishes its NDOs to Rendezvous and when another node subscribes to this NDO, the publication and Subscription are matched by the Rendezvous module. If there is a tree for the sub/pub, then data transfer starts straight away, otherwise the forwarding module will forward data based on the labels of the packets.

III. Overview of LISP

To provide improved routing scalability while facilitating flexible address assignment for multi-homing and mobility, the LISP describes changes to the Internet architecture in which IP addresses are replaced by Routing Locators (RLOCs) for routing through the global Internet and by Endpoint Identifiers (EIDs) for identifying network sessions between devices [10]. As shown in Fig 3, three essential components exist in an LISP environment: LISP sites (EID space), non-LISP sites (RLOC space), and LISP Mapping System which comprises Map Servers (MS) and databases.

- **LISP sites (EID space):** They represent customer end-sites in exactly the same way that end-sites are defined today. However, the IP address in the EID space are not advertised to the non-LISP sites, but are published into the LISP Mapping Systems which perform the EID-RLOC mapping. The SLIP functionalities is deployed on the site's gateway or edge routers. Therefore, based on their roles, two types of the router are defined: Firstly, the Ingress Tunnel Routers (ITRs) which receive packets from site end-systems and sends LISP-encapsulated packets toward the Internet. Secondly, the Egress Tunnel Routers (ETRs) which receive LISP-encapsulated packets from the Internet and sends de-capsulated packets to site end-systems [10], [11].
- **Non-LISP sites (RLOC space):** They represent current sites where the IP addresses are advertised and used for routing purpose.
- **LISP Mapping Systems:** These are represented by a globally distributed database that contains all known EID prefixes to RLOC mappings and Map Servers (MS). Similar to the current Domain Name System (DNS), the Mapping systems are queried by LISP-capable devices for EID-To-RLOC mapping.

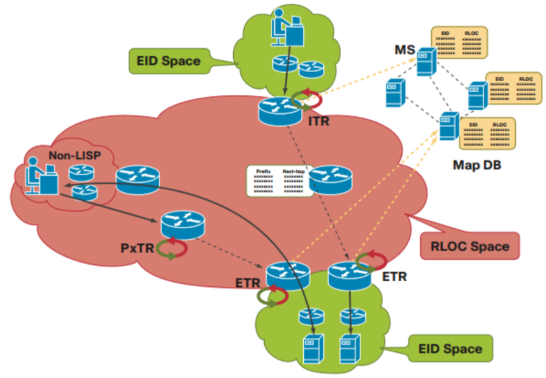

Figure 3. The LISP Network Architecture Design [10]

A. *Interactions With Other LISP Components*

The functionality of the LISP architecture comprises two stages:

1) EID Prefix Configuration and ETR Registration:: As explained in [12], an ETR publishes its EID-prefixes on a Map Server (MS) by sending LISP Map-Register messages which includes the ETR's RLOC and a list of its EID-prefixes. Upon receipt of a Map-Register from an ETR, the Map Server checks the validity of the Map-Register message and acknowledges it by sending a Map-Notify message.

2) The Address Resolving Stage:: Once a Map Server has EID-prefixes registered by its client ITRs, it will accept and process Map-Requests from them. In response to a Map-Request (sent from an ITR), the Map Server first checks to see if the destination EID matches a configured EID-prefix. If there is no match, the Map Server returns a negative Map-Reply message to the ITR. In case of a match,

the Map Server might operate in either of two modes:

1) **No-Proxy Reply Service:** If none of the registered ITRs have requested proxy reply service, then the Map Server forwards the received Map-Request to one of the registered ETRs which will then respond to the ITR by returning a Map-Reply message.
2) **Proxy Reply Service:** If any of the registered ITRs for the EID-prefix have requested proxy reply service, then the Map Server, upon receiving a Map-Request, answers the request instead of forwarding it to the corresponding ETR.

More deatiled about these modes could be found in [12]. Furthermore, the LISP working group in [11] has defined the structure of all the LISP packets including the Map-Request, the Map-Notify, the Map-Register and the MAP-Reply.

IV. The Proposed Model

This model is based on the network structure as shown in Fig 3, and it comprises two stages. The Subscription Stage,

Table I
THE ITR DATABASE

The EID	NDO
EID1	NDO1
	NDO3
	NDO7
EID5	NDO1
	NDO3
	NDO7

Table II
THE MS DATABASE

NDO	RLOC	Weight
NDO1	RLOC1	w11
	RLOC3	w12
	RLOC5	w13
NDO6	RLOC2	w61
	RLOC5	w62
	RLOC9	w63

where nodes register the NDOs with the mapping system. The Forwarding Stage, where information is brought to the requesting node from the closest source.

A. The Subscription Stage

Each node (called publisher) registers its NDOs with the local ITR, the ITR publishes the registered NDOs into the mapping system using the LISP Register packet which includes the NDOs and the ITR's RLOC. The Map Server acknowledges this with the LISP Notify packet. At the end of this stage, the ITR will have a table of all the EIDs and their NDOs as in Table I. Also, the MS will have a database of all the NDOs, the corresponding RLOCs and the weight as shown in Table II. The weight field will be discussed in section V.

B. The Forwarding Stage

In this stage, we differentiate between two cases. The initial case is when a specific NDO is requested for the first time (it has never been requested before). The second case deals with the forwarding of a NDO that has been previously requested.

- **The Initial Case:** As shown in Fig 4, when a node (the Requester) requests a specific NDO, it sends the request to the ITR (message 1), which composes an LISP Map-Request packet with the requested NDO and the ITR's RLOC towards the MS (message 2). Upon receipt of the Map-Request, the MS will look up the database, and here the MS might react in two ways:
  1) **Proxy Server:** During the Registration stage, an ITR might ask the MS to act as a proxy. In this case, when receiving the Map-Request, the MS looks up the database, finds the correspondent ETR that has the NDO and encapsulates the request towards it (message 3). The ETR decapsulates the request and gets the required NDO from the source node in its network (message 4). The ETR will respond directly to the ITR using the LISP Map-Reply packet which includes the requested NDO (not shown in the figure).
  2) **No Proxy Server:** In this case, the MS will not forward the Map-Request to the relevant ITR, rather it returns the Map-Reply packet which indicates the ETR as the source of information (message 3). In (message 4), the ITR directly approaches the ETR, which will get the NDO from the source (message 5) and then respond directly to the ITR (not shown in the figure).

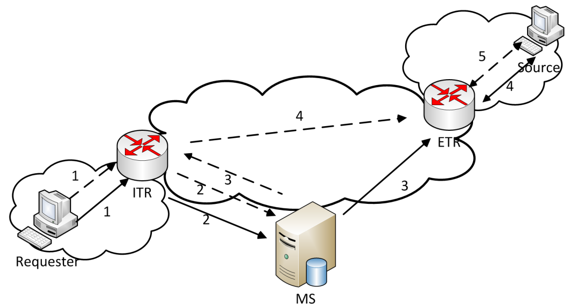

Figure 4. Proxy-Enabled (solid arrows).No Proxy-Enabled (dashed arrows)

- **The Following Case:**

The content of the ETRs' responds that include the NDOs will be cached by the ITR, ETR and some intermediate LISP-capable routers in the path between the ITR and ETR. The criteria of which routers to cache the NDO is based on the NDO's "scope" which should be defined by the publisher. However, this is beyond the scope of this paper and will be discussed in future work. Each router which caches NDOs will periodically report this to the MS via the LISP Register packet. So when a request for a specific NDO comes to the MS, it will check the data base and get the NDO from the nearest source to the requester. More details about the caching and forwarding procedures are in the following section.

V. THE CACHING AND FORWARDING PROCEDURES

A. Caching Criteria

We presume that each LISP-capable router has a fixed-size cache, and each NDO will initially be cached for a predefined period of time (t). When an NDO is cached by a router, a timer will start to count down and when it reaches zero, the NDO will be released from the cache. The value of the timer exponentially increases when the same NDO is cached again by the same router, this way the most frequently requested data will be cached for a longer period of time. Furthermore, whenever a cached NDO is requested, the timer resets. The router should be able to keep records of how many time each timer has been rest (n), and for each cached NDO, a weight value (w) is computed as follows: $w = f(t, n)$, the actual function (f) is not defined in this paper as it is related to routers' cache size; therefore, it will be defined later on as part of the implementation stage of our project.

The cache's contents are ordered based on the weight, hence the NDOs with small weight will be timed out sooner

Figure 5. The Grid in the Map Server

and are most likely to be released form the cache. In the case when the cache is full and none of the NDOs are timed out, the router will not be able to cache any new NDOs.

B. The Forwarding Procedure

Periodically (every time interval δ), the LISP-capable routers updates the MS with a list of the cached NDOs along with the corresponding weight. For each NDO, the MS will keep a value (λ) which is the time of the next update, this is defined as the time of the last update (T_1) plus the time interval (δ) that defines how often an update is sent to the MS: $\lambda = (T_1 + \delta)$. The MS will maintain the information about all the routers in a grid format as in Fig 5. When a request for a specific NDO comes to the MS, it needs to launch the data from the nearest caching router to the requester. For example, if a request comes from the router (X1/Y1) and the requested NDO is available at routers (X2/Y3 & X4/Y4), then the MS will define the nearest source of data by computing the distance (d) between the requesting router and the possible sources as follows: $d1 = \sqrt{(X2 - X1)^2 + (Y3 - Y1)^2}, d2 = \sqrt{(X4 - X1)^2 + (Y4 - Y1)^2}$.

The distance metric is not enough on its own; because since routers use fixed-size caches, their cached data might change frequently. It might be the case that the information about an NDO available at the MS is outdated, this might happen if the cache content changes straight after the router updates the MS. To deal with this issue, the forwarding procedure must consider one more metric. Therefore, when a request for an NDO comes to the MS, it checks the weight (w) of the NDO, compares it with the value Λ of the possible sources. As shown in Fig 6, for a specific NDO, if the $\lambda > w$, the required NDO will not be available at the router. Therefore, the MS will check if the NDO is cached in another router. If $\lambda < w$, the distance metric will be checked as explained in Fig 6.

C. Current Work

Work has begun at the Network Research Laboratory, Middlesex University on developing a testbed which will be used to test the proposed model. The testbed will first develop the network-level communication using LISP-capable router (Cisco Nexus 7000 Series). Furthermore, using the Blade Server at our lab, we will set up a virtual environment that comprises a distributed database acting as a Map Server, and a number of file servers. This will allow us to test the proposed model. At a later stage, we will add caching capabilities at each node to see how efficient is the system.

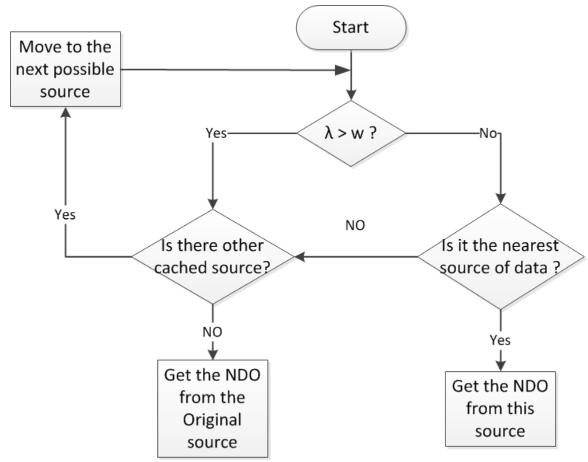

Figure 6. The Forwarding Algorithm

VI. Conclusion

In order to support host-independent routing as well as an in-network caching for data in Information Centric Networks, the paper proposes a new approach based on the LISP network architecture to separate the roles of identifier and locator. The proposal realised the need for new caching and forwarding procedures to cope with the large volume of exchanged data in the ICNs. The aim of this paper is to describe our new vision of using LISP as convergence layer protocol to support communication in ICNs. Currently, we are working on building a mathematical model that represent our system. Furthermore, work has begun at the Network Research Laboratory, Middlesex University on developing a testbed which will be used to test the proposed model. The testbed will first develop the network-level communication using LISP-capable router (Cisco Nexus 7000 Series). Furthermore, using the Blade Server at our lab, we will set up a virtual environment that comprises a distributed database acting as a Map Server, and a number of file servers. This will allow us to test the proposed model. At a later stage, we will add caching capabilities at each node to see how efficient the system is.

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] T. Koponen, A. Ermolinskiy, M. Chawla, K.H. Kim, I. Stoica, B. Chun and S. Shenker, *A data-oriented (and beyond) network architecture*, In SIGCOMM, 2007.

[3] D.h. Kim and J.h. Kim and Y.s. Kim and H.s. Yoon and I. Yeom, *Mobility Support in Content Centric Networks*, In SIGCOMM, 2012.

[4] M. Sarela, T. Rinta Aho, T. Tarkoma, *RTFM: Publish/subscribe internetworking architecture*, In ICT Mobile Summit , 2008.

[5] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher , B. Ohlman, *A survey of information-centric networking*, IEEE Communication Magazine, Volume: 50 , Issue: 7 , 2012.

[6] W.K. Chai and D. He and I. Psaras and G. Pavlou, *Cache "Less for More" in Information-Centric Networks*, In Information-Centric Networks. Networking, 2012.

[7] Y. Rekhter, and T. Li and S. Hares, *A Border Gateway Protocol 4 (BGP-4)*,Request for Comments: 4271, 2006.

[8] A.R. Atkinson. *ILNP Concept of Operation*. internet Draft, 2008.

[9] G. Mapp , M. Aiash , H.C. Guardia , J. Crowcroft, *Exploring Multi-homing Issues in Heterogeneous Environments*. Advanced Information Networking and Applications (WAINA), 2011.

[10] Cisco Locator/ID Separation Protocol Revolutionary Network Architecture to Power the Network. http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/aag_c45-635298.pdf. [Last Accessed on 13.01.13].

[11] D. Farinacci, V. Fuller, D. Meyer, D. Lewis,*Locator/ID Separation Protocol (LISP)*. Internet-Draft , November 13, 2012.

[12] D. Farinacci,V. Fuller, *LISP Map Server Interface*. Internet-Draft , March 4, 2012.

[13] Cisco Nexus 7000 Series Switches, Cisco Nexus 7000 LISP Overview Video: http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6554/ps6599/ps10800/LISP_VDS.html. [Last Accessed on 13.01.13].